NAG Toolbox for MATLAB

f02wu

1 Purpose

f02wu returns all, or part, of the singular value decomposition of a real upper triangular matrix.

2 Syntax

```
[a, b, q, sv, work, ifail] = f02wu(a, b, wantq, wantp, 'n', n, 'ncolb',
ncolb)
```

3 Description

The n by n upper triangular matrix R is factorized as

$$R = QSP^{\mathrm{T}},$$

where Q and P are n by n orthogonal matrices and S is an n by n diagonal matrix with nonnegative diagonal elements, $\sigma_1, \sigma_2, \ldots, \sigma_n$, ordered such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0.$$

The columns of Q are the left-hand singular vectors of R, the diagonal elements of S are the singular values of R and the columns of P are the right-hand singular vectors of R.

Either or both of Q and P^{T} may be requested and the matrix C given by

$$C = Q^{\mathrm{T}}B,$$

where B is an n by ncolb given matrix, may also be requested.

The function obtains the singular value decomposition by first reducing R to bidiagonal form by means of Givens plane rotations and then using the QR algorithm to obtain the singular value decomposition of the bidiagonal form.

Good background descriptions to the singular value decomposition are given in Chan 1982, Dongarra *et al.* 1979, Golub and Van Loan 1996, Hammarling 1985 and Wilkinson 1978.

Note that if K is any orthogonal diagonal matrix so that

$$KK^{\mathrm{T}} = I$$

(that is the diagonal elements of K are $+1$ or -1) then

$$A = (QK)S(PK)^{\mathrm{T}}$$

is also a singular value decomposition of A.

4 References

Chan T F 1982 An improved algorithm for computing the singular value decomposition *ACM Trans. Math. Software* **8** 72–83

Dongarra J J, Moler C B, Bunch J R and Stewart G W 1979 *LINPACK Users' Guide* SIAM, Philadelphia

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Hammarling S 1985 The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

Wilkinson J H 1978 Singular Value Decomposition – Basic Aspects *Numerical Software – Needs and Availability* (ed D A H Jacobs) Academic Press

5 Parameters

5.1 Compulsory Input Parameters

1: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1,\mathbf{n})$

The second dimension of the array must be at least $\max(1,\mathbf{n})$

The leading n by n upper triangular part of the array **a** must contain the upper triangular matrix R.

2: **b(ldb,$*$) – double array**

The first dimension, **ldb**, of the array **b** must satisfy

if $\mathbf{ncolb} > 0$, $\mathbf{ldb} \geq \max(1,\mathbf{n})$;
$\mathbf{ldb} \geq 1$ otherwise.

The second dimension of the array must be at least $\max(1,\mathbf{ncolb})$

With $\mathbf{ncolb} > 0$, the leading n by $ncolb$ part of the array **b** must contain the matrix to be transformed.

3: **wantq – logical scalar**

Must be **true** if the matrix Q is required.

If **wantq** $=$ **false**, the array **q** is not referenced.

4: **wantp – logical scalar**

Must be **true** if the matrix P^{T} is required, in which case P^{T} is overwritten on the array **a**, otherwise **wantp** must be **false**.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a**.

n, the order of the matrix R.

If $\mathbf{n} = 0$, an immediate return is effected.

Constraint: $\mathbf{n} \geq 0$.

2: **ncolb – int32 scalar**

Default: The second dimension of the array **b**.

$ncolb$, the number of columns of the matrix B.

If $\mathbf{ncolb} = 0$, the array **b** is not referenced.

Constraint: $\mathbf{ncolb} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldq

5.4 Output Parameters

1: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1,\mathbf{n})$

The second dimension of the array must be at least $\max(1,\mathbf{n})$

If **wantp** $=$ **true**, the n by n part of **a** will contain the n by n orthogonal matrix P^{T}, otherwise the n by n upper triangular part of **a** is used as internal workspace, but the strictly lower triangular part of **a** is not referenced.

2: **b(ldb,$*$) – double array**

The first dimension, **ldb**, of the array **b** must satisfy

if **ncolb** > 0, **ldb** $\geq \max(1, \mathbf{n})$;
ldb ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{ncolb})$

The leading n by $ncolb$ part of the array **b** contains the matrix $\mathbf{q}^{\mathrm{T}}\mathbf{b}$.

3: **q(ldq,$*$) – double array**

The first dimension, **ldq**, of the array **q** must satisfy

if **wantq** $=$ **true**, **ldq** $\geq \max(1, \mathbf{n})$;
ldq ≥ 1 otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **wantq** $=$ **true**, and at least 1 otherwise

With **wantq** $=$ **true**, the leading n by n part of the array **q** will contain the orthogonal matrix Q. Otherwise the array **q** is not referenced.

4: **sv($*$) – double array**

Note: the dimension of the array **sv** must be at least $\max(1, \mathbf{n})$.

The array **sv** will contain the n diagonal elements of the matrix S.

5: **work($*$) – double array**

Note: the dimension of the array **work** must be at least $\max(1, 2 \times (\mathbf{n} - 1))$ if **ncolb** $= 0$ and **wantq** $=$ **false** and **wantp** $=$ **false**, $\max(1, 3 \times (\mathbf{n} - 1))$ if (**ncolb** $= 0$ and **wantq** $=$ **false** and **wantp** $=$ **true**) or (**wantp** $=$ **false** and (**ncolb** > 0 or **wantq** $=$ **true**)), and at least $\max(1, 5 \times (\mathbf{n} - 1))$ otherwise.

work(n) contains the total number of iterations taken by the QR algorithm.

The rest of the array is used as internal workspace.

6: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= -1$

On entry, **n** < 0,
or **lda** $<$ **n**,
or **ncolb** < 0,
or **ldb** $<$ **n** and **ncolb** > 0,
or **ldq** $<$ **n** and **wantq** $=$ **true**.

ifail > 0

The QR algorithm has failed to converge in $50 \times \mathbf{n}$ iterations. In this case $\mathbf{sv}(1), \mathbf{sv}(2), \ldots, \mathbf{sv}(\mathbf{ifail})$ may not have been found correctly and the remaining singular values may not be the smallest. The matrix R will nevertheless have been factorized as $R = QEP^{\mathrm{T}}$, where E is a bidiagonal matrix with $\mathbf{sv}(1), \mathbf{sv}(2), \ldots, \mathbf{sv}(n)$ as the diagonal elements and $\mathbf{work}(1), \mathbf{work}(2), \ldots, \mathbf{work}(n-1)$ as the superdiagonal elements.

This failure is not likely to occur.

7 Accuracy

The computed factors **q**, S and P satisfy the relation

$$QSP^{\mathrm{T}} = R + E,$$

where

$$\|E\| \leq c\epsilon\|A\|,$$

ϵ is the ***machine precision***, c is a modest function of n and $\|.\|$ denotes the spectral (two) norm. Note that $\|A\| = \mathbf{sv}(1)$.

A similar result holds for the computed matrix $Q^{\mathrm{T}}B$.

The computed matrix Q satisfies the relation

$$Q = T + F,$$

where T is exactly orthogonal and

$$\|F\| \leq d\epsilon,$$

where d is a modest function of n. A similar result holds for P.

8 Further Comments

For given values of **ncolb**, **wantq** and **wantp**, the number of floating-point operations required is approximately proportional to n^3.

>Following the use of f02wu the rank of R may be estimated as follows:

```
tol = eps;
irank = 1;
while (irank <= numel(sv) && sv(irank) >= tol*sv(1) )
  irank = irank + 1;
end
```

returns the value k in `irank`, where k is the smallest integer for which $\mathbf{sv}(k) < tol \times \mathbf{sv}(1)$, where tol is typically the machine precision, so that `irank` is an estimate of the rank of S and thus also of R.

9 Example

```
a = [-4, -2, -3;
      0, -3, -2;
      0, 0, -4];
b = [-1; -1; -1];
wantq = true;
wantp = true;
[aOut, bOut, q, sv, work, ifail] = f02wu(a, b, wantq, wantp)
```

```
aOut =
    0.4694    0.4324    0.7699
   -0.7845   -0.1961    0.5883
    0.4054   -0.8801    0.2471
bOut =
    1.6716
    0.3922
   -0.2276
q =
```

```
   -0.7699    0.5883   -0.2471
   -0.4324   -0.1961    0.8801
   -0.4694   -0.7845   -0.4054
sv =
    6.5616
    3.0000
    2.4384
work =
   -0.0000
    0.0000
    5.0000
    2.2500
    3.2500
    0.0000
    1.0000
    1.0000
    0.0008
    0.0000
ifail =
           0
```